

Abhandlungen

Ein Beitrag zur Numerik von Polynomen und Polynommatrizen

SIGURD FALK

Echternstr. 46, D-38100 Braunschweig

Zusammenfassung

Es werden einige auch für mehrfache Nullstellen bzw. Eigenwerte quadratisch konvergierende Algorithmen bereitgestellt. Dabei dienen die von MATLAB ausgeworfenen Näherungen als Startwerte. Acht numerische Beispiele demonstrieren die Zuverlässigkeit dieser Vorgehensweise.

1. Aufgabenstellung

Vorgegeben ist die Matrizen-Eigenwertgleichung

$$\mathbf{y}^T \mathbf{F}(\lambda) = \mathbf{o}^T; \quad \mathbf{F}(\lambda) \mathbf{x} = \mathbf{o} \quad (1.1)$$

mit der Polynommatrix der Ordnung n vom Grade p

$$\mathbf{F}(\lambda) = A_0 + A_1 \lambda + A_2 \lambda^2 + \dots + A_p \lambda^p; \quad \det A_p \neq 0. \quad (1.2)$$

mit im allgemeinen komplexwertigen Koeffizientenmatrizen A_0 bis A_p .
Ihre Eigenwerte

$$\lambda_1, \lambda_2, \dots, \lambda_m; \quad m = p \cdot n \quad (1.3)$$

werden im Folgenden als Nullstellen des charakteristischen Polynoms

$$\det \mathbf{F}(\lambda) = f(\lambda) = a_0 + a_1 \lambda + a_2 \lambda^2 + \dots + a_m \lambda^m \quad (1.4)$$

berechnet mittels einer Padé-Funktion¹

Eingegangen: 27.04.2013.

¹ Henri Eugene Padé, französischer Mathematiker 1863–1953

$$p(\lambda) = \frac{f(\lambda)}{z(\lambda)}, \quad (1.5)$$

wo $z(\lambda)$ ein Polynom vom Grad $\leq m$ ist.

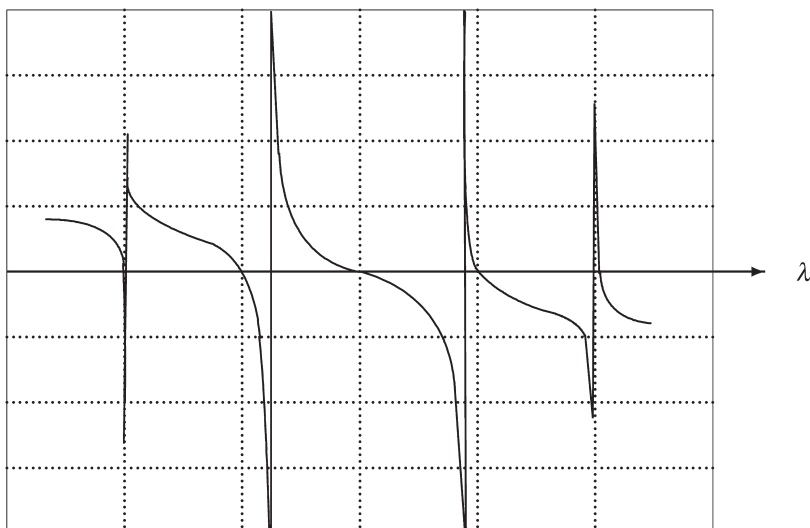


Bild 1: Typischer Verlauf einer Padé-Funktion im Reellen.

Die spezielle Padé-Funktion mit $z(\lambda) = -f'(\lambda)$, somit

$$p(\lambda) = \frac{f(\lambda)}{-f'(\lambda)} \quad (1.6)$$

zeichnet sich durch die folgende Eigenschaft aus. Das Polynom $f(\lambda)$ besitze eine Nullstelle a der Vielfachheit v , damit gilt die Darstellung

$$f(\lambda) = (\lambda - a)^v \cdot z(\lambda); \quad z(a) \neq 0. \quad (1.7)$$

Es folgt

$$f'(\lambda) = v(\lambda - a)^{v-1} \cdot z(\lambda) + (\lambda - a)^v \cdot z'(\lambda), \quad (1.8)$$

oder auch

$$f'(\lambda) = (\lambda - a)^{v-1} \left[\underbrace{v \cdot z(\lambda) + (\lambda - a) \cdot z'(\lambda)}_{\varphi(\lambda)} \right], \quad (1.9)$$

wo

$$\varphi(a) = v \cdot z(a) + 0 \neq 0 \quad (1.10)$$

ist.

Damit wird die Padé-Funktion

$$p(\lambda) = \frac{f(\lambda)}{-f'(\lambda)} = \frac{(\lambda - a)^v \cdot z(\lambda)}{-(\lambda - a)^{v-1} \cdot \varphi(\lambda)} = (\lambda - a) \cdot \frac{z(\lambda)}{-\varphi(\lambda)}. \quad (1.11)$$

Es gilt demnach die **Aussage**:

Die Padé-Funktion (1.6) besitzt ausschließlich einfache Nullstellen.

2. Die Algorithmen

Für den Anwender kommen nur solche Algorithmen in Betracht, die auch bei mehrfachen Nullstellen sicher zum Ziel führen. Das auf Newton zurückgehende Tangentenverfahren scheidet daher von vornherein aus. In praxi bewährt haben sich dagegen die drei folgenden Typen.

2 a) Grundlage ist die Padé-Funktion (1.6). Damit wird

$$\Lambda_{j+1} = \Lambda_j + p(\Lambda_j); \quad j = 1, 2, \dots \quad (2.1)$$

2 b) Grundlage ist die nach Halley benannte Funktion

$$h(\lambda) = \frac{p(\lambda)}{1 + p(\lambda) \cdot q(\lambda)}, \quad (2.2)$$

wo

$$q(\lambda) = \frac{f''(\lambda)}{f'(\lambda)} \quad (2.3)$$

ist. Damit wird

$$\Lambda_{j+1} = \Lambda_j + h(\Lambda_j); \quad j = 1, 2, \dots \quad (2.4)$$

2 c) Grundlage ist die Familie von Testpolynomen

$$f_k(\lambda) = a_0 + (-1)^k \cdot a_2 \lambda^2 + (-2)^k \cdot a_3 \lambda^3 + \dots + (1-m)^k \cdot a_m \lambda^m \quad (2.5)$$

mit

$$k = 1, 2, \dots, v, \quad (2.6)$$

wo v die Vielfachheit der angesteuerten Nullstelle ist.

Es besitze das vorgegebene Polynom $f_0(\lambda)$ eine Nullstelle $\tilde{\lambda}$ der Vielfachheit v , dann besitzt jede der Padé-Funktionen

$$P_1(\lambda) = \frac{f_0(\lambda)}{f_1(\lambda)}; \quad P_2(\lambda) = \frac{f_1(\lambda)}{f_2(\lambda)}, \quad \dots \quad P_v(\lambda) = \frac{f_{v-1}(\lambda)}{f_v(\lambda)} \quad (2.7)$$

diese als einfache Nullstelle. Mit

$$p_v(\lambda) = P_v(\lambda) \cdot \lambda \quad (2.8)$$

lautet der Algorithmus

$$\Lambda_{j+1} = \Lambda_j + p_v(\Lambda_j); \quad j = 1, 2, \dots \quad (2.9)$$

oder auch

$$\Lambda_{j+1} = [1 + P_v(\Lambda_j)] \cdot \Lambda_j; \quad j = 1, 2, \dots \quad (2.10)$$

Alle drei Algorithmen konvergieren quadratisch.

3. Sondierung

Die Sondierung dient der Bereitstellung von Näherungswerten, mit denen die Algorithmen gestartet werden.

Hier sind drei Fälle zu unterscheiden.

3 a) Die Koeffizienten des vorgegebenen Polynoms (1.4) sind ebenso wie die Nullstellen reell. Grundlage ist die Padé-Funktion $p(\lambda)$ (1.6).

3 a1) Die herkömmliche Regula falsi.

Für irgend zwei Testpunkte λ_1 und λ_2 besteht ein Vorzeichenwechsel, gekennzeichnet durch

$$\lambda_1 < \lambda_2; \quad p(\lambda_1) > 0, \quad p(\lambda_2) < 0. \quad (3.1)$$

Dann liegt zwischen λ_1 und λ_2 eine Nullstelle der Funktion $p(\lambda)$ und damit auch eine (womöglich mehrfache) Nullstelle von $f(\lambda)$.

Die Regula falsi mit dem Differenzenquotienten

$$\Delta_2 = \frac{p(\lambda_2) - p(\lambda_1)}{\lambda_2 - \lambda_1} \quad (3.2)$$

liefert eine erste (im allgemeinen recht grobe) Näherung

$$\lambda_3 = \lambda_1 - \frac{p(\lambda_1)}{\Delta_2}. \quad (3.3)$$

3 a2) Die beschleunigte Regula falsi

Die Näherung λ_3 lässt sich verbessern durch Hinzunahme eines weiteren Differenzenquotienten

$$\Delta_3 = \frac{p(\lambda_3) - p(\lambda_1)}{\lambda_3 - \lambda_1} \quad (3.4)$$

sowie den beiden Quotienten

$$Q_2 = \frac{p(\lambda_2)}{p(\lambda_1)}; \quad Q_3 = \frac{p(\lambda_3)}{p(\lambda_1)}. \quad (3.5)$$

Damit wird

$$\lambda_4 = \lambda_1 = \frac{p(\lambda_2) - p(\lambda_3)}{Q_2\Delta_3 - Q_3\Delta_2} \quad (3.6)$$

und dies lässt sich iterativ fortsetzen auf folgende Weise.

Erster Schritt.

Ersetze in (3.2) bis (3.6) die Indizes 1,2 und 3 durch 2,3 und 4, und berechne damit das Wertepaar λ_5 ; $p(\lambda_5)$.

Zweiter Schritt.

Ersetze in (3.2) bis (3.6) die Indizes 1,2 und 3 durch 3,4 und 5, und berechne damit das Wertepaar λ_6 ; $p(\lambda_6)$.

Diese damit erklärte Iteration wird abgebrochen, wenn a)

$$|p(\lambda_\mu)| \leq 10^{-\sigma} \quad (3.7)$$

geworden ist oder

b) unter Verzicht auf ein Abbruchkriterium ein Index μ vorgegeben wird.

Nun zur Sondierung. Nach Wahl einer Schrittweite δ werden auf der λ -Achse von Null beginnend die Wertepaare

$$\lambda_j; p(\lambda_j); \quad j = 1, 2, \dots \quad (3.8)$$

berechnet bis ein erster, zweiter, dritter usw. Vorzeichenwechsel gefunden wurde.

3 b) Um auch die negativen Nullstellen einzuschließen, wird mit der Ko-Funktion

$$\hat{p}(\lambda) = \frac{f(-\lambda)}{-f'(-\lambda)} \quad (3.9)$$

gleichzeitig die Folge (3.4) durchgeführt bis alle m Nullstellen ermittelt sind.

3 c) Sehr viel aufwendiger gestaltet sich die Sondierung innerhalb der komplexen Zahlenebene, da hier kein Vorzeichenwechsel nach Art von (3.1) zur Verfügung steht.

3 d) Diagonaldominante Polynommatrizen

Bei hinreichend ausgeprägter Diagonaldominanz sind die $m = \rho \cdot n$ Nullstellen der Gleichungen

$$f_{jj}(\lambda) = 0; \quad j = 1, 2, \dots, n \quad (3.10)$$

brauchbare Startwerte für die im Abschnitt 2 aufgeführten Algorithmen. Im Sonderfall $\rho = 2$ sind somit n quadratische Gleichungen zu lösen, siehe dazu auch das zweite Beispiel.

4. Eigenvektoren einer Polynommatrix

Es geht im Folgenden um die Gleichungen (1.1)

$$\mathbf{F}(\lambda)\mathbf{x} = \mathbf{0} \quad (4.1)$$

mit der Polynommatrix (1.2).

Es sei λ_k ein einfacher Eigenwert, dann besitzt die Matrix

$$\mathbf{F}(\lambda_k) \quad (4.2)$$

den Rang $n-1$.

a) Transformation nach Gauß

Spaltenpivotsuche in der Reihenfolge 1, 2, ..., m-1 und Vertauschung zweier Zeilen, sofern dies erforderlich wird. Auf diese Weise geht die Matrix (4.2) über in

$$\tilde{\mathbf{F}}(\lambda_k) = \begin{bmatrix} \tilde{\nabla}_k & \mathbf{w}_k \\ \mathbf{o}^T & 0 \end{bmatrix} \quad (4.3)$$

mit einer regulären oberen Dreiecksmatrix $\tilde{\nabla}_k$ der Ordnung n-1 und einer Spalte \mathbf{w}_k der Länge n-1.

b) Transformation nach Jordan² in der Reihenfolge

$$n-1, \quad n-2, \quad \dots \quad 2. \quad (4.4)$$

Damit geht die Matrix (4.3) über in

$$\hat{\mathbf{F}}(\lambda_k) = \begin{bmatrix} \mathbf{D}_k & \mathbf{z}_k \\ \mathbf{o}^T & 0 \end{bmatrix} \quad (4.5)$$

mit einer regulären Diagonalmatrix \mathbf{D}_k der Ordnung n-1.

Der gesuchte Eigenvektor ist dann

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{D}_k^{-1} \mathbf{z}_k \\ -1 \end{bmatrix}, \quad (4.6)$$

wovon man sich leicht überzeugt.

Da das Gleichungssystem (1.1) homogen ist, ist auch

$$\mathbf{x}_k = \mathbf{x}_k \cdot \alpha_k; \quad \alpha_k \neq 0 \quad (4.7)$$

ein Eigenvektor, wo der Faktor α_k geeignet zu wählen ist, etwa zum Zweck der Normierung gemäß

$$\mathbf{x}_k^* \mathbf{x}_k = 1, \quad (4.8)$$

doch wird man im allgemeinen

$$\alpha_k = 1 \quad (4.9)$$

setzen.

² Wilhelm Jordan, Geometer 1842–1899

Mehrfache Eigenwerte

Es sei λ_k eine Nullstelle der Vielfachheit v_k und r_k der Rangabfall der Matrix $\mathbf{F}(\lambda_k)$, wo bekanntlich

$$r_k \leq v_k \quad (4.10)$$

gilt.

Die Matrix (4.3) ist jetzt allgemeiner als dort

$$\tilde{\mathbf{F}}(\lambda_k) = \begin{bmatrix} \tilde{\nabla} & \mathbf{0} \\ \mathbf{0}^T & \mathbf{0} \end{bmatrix}, \quad (4.11)$$

wo die Nullmatrix unten rechts die Ordnung r_k besitzt.

Es folgt analog zu (4.3) bis (4.6)

$$\hat{\mathbf{F}}(\lambda_k) = \begin{bmatrix} \mathbf{D}_k & \tilde{\mathbf{Z}}_k \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (4.12)$$

und daraus

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{D}_k^{-1} \tilde{\mathbf{Z}}_k \\ -\mathbf{I}_{r_k} \end{bmatrix} = (\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_{r_k}), \quad (4.13)$$

und dies sind r_k linear unabhängige Eigenvektoren zum Eigenwert λ_k , wo jeder von ihnen nach (4.7) bis (4.9) normiert werden kann.

Falls $r_k < v_k$ ist, können bei Bedarf die r_k Eigenvektoren (4.13) durch Hauptvektoren ergänzt werden, siehe dazu [10, Seite 253].

Nun zu den Linkseigenvektoren. Aus der Gleichung

$$\mathbf{y}^T \mathbf{F}(\lambda) = \mathbf{0} \quad (4.14)$$

folgt durch Transposition

$$[\mathbf{y}^T \mathbf{F}(\lambda)]^T = [\mathbf{0}^T]^T \rightarrow \mathbf{F}^T(\lambda) \mathbf{y} = \mathbf{0}, \quad (4.15)$$

und damit kann alles vorige sinngemäß übernommen werden.

5. Die ECP-Transformation

Dieser in [10, S. 411] beschriebene Algorithmus basiert auf der Bereitstellung von m paarweise verschiedenen (disjunkten) Stützwerten

$$\sigma_1, \sigma_2, \dots, \sigma_m, \quad (5.1)$$

die geeignet zu wählen sind. Mit diesen sind definiert die Stützpolynome

$$g_k(\lambda) = \sum_{\substack{j=1 \\ j \neq k}}^m (\sigma_j - \lambda); \quad k = 1, 2, \dots, m \quad (5.2)$$

und damit die Padé-Funktionen

$$P_k(\lambda) = \frac{f(\lambda)}{g_k(\lambda)} \cdot \frac{1}{a_m}; \quad k = 1, 2, \dots, m. \quad (5.3)$$

Für $\lambda = \sigma_k$ entsteht daraus der (in [1] und [2] auch als Knoten bezeichnete) Defekt

$$d_k = \sigma_k - d_k; \quad k = 1, 2, \dots, m. \quad (5.4)$$

und damit der dem Defekt zugeordnete sogenannte **Hauptwert**

$$H_k = \frac{f(\sigma_k)}{g_k(\sigma_k)} \cdot \frac{1}{a_m}; \quad k = 1, 2, \dots, m \quad (5.5)$$

Alle diese Werte werden zusammengestellt in einer Liste

$$L_m = \left\{ \begin{array}{ccc} \text{Stützwerte} & \text{Defekte} & \text{Hauptwerte} \\ \hline \sigma_1 & d_1 & H_1 \\ \sigma_2 & d_2 & H_2 \\ \vdots & \vdots & \vdots \\ \sigma_m & d_m & H_m \end{array} \right\}, \quad (5.6)$$

welche die gesamte Information der Polynommatrix (1.2) enthält. Die Reihenfolge der einzelnen Zeilen ist dabei beliebig. Die Kontrollgleichung

$$\sum_{j=1}^m H_j = -\frac{a_{m-1}}{a_m}, \quad (5.7)$$

verbürgt die fehlerfreie Berechnung der Defekte (5.4) aus den Stützwerten (5.1).

6. Der ECP-Rayleigh-Quotient

Nach [3, Seite 42] gilt die Eigenwertgleichung

$$\det \mathbf{F}(\lambda) = \det(\mathbf{E} - \lambda \mathbf{I}_m) = 0 \quad (6.1)$$

mit der ECP-Begleitmatrix

$$\mathbf{E} = \mathbf{Diag} \langle \sigma_j \rangle - \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} (d_1 \quad d_2 \dots d_m). \quad (6.2)$$

Der mit dem Vektorpaar

$$\mathbf{y}^T(\lambda) = \begin{bmatrix} \frac{1}{\sigma_1 - \lambda} & \frac{1}{\sigma_2 - \lambda} & \dots & \frac{1}{\sigma_m - \lambda} \end{bmatrix}; \quad \mathbf{x}(\lambda) = \begin{bmatrix} \frac{d_1}{\sigma_1 - \lambda} \\ \frac{d_2}{\sigma_2 - \lambda} \\ \vdots \\ \frac{d_m}{\sigma_m - \lambda} \end{bmatrix} \quad (6.3)$$

gebildete Rayleigh-Quotient

$$R(\lambda) = \frac{\mathbf{y}^T(\lambda) \mathbf{E} \mathbf{x}(\lambda)}{\mathbf{y}^T(\lambda) \mathbf{I}_m \mathbf{x}(\lambda)} \quad (6.4)$$

führt mit den Summen

$$S_1(\lambda) = \sum_{j=1}^m \frac{d_j}{\sigma_j - \lambda}, \quad (6.5)$$

$$S_2(\lambda) = \sum_{j=1}^m \frac{d_j}{(\sigma_j - \lambda)^2}, \quad (6.6)$$

$$S_\sigma(\lambda) = \sum_{j=1}^m \frac{d_j \sigma_j}{(\sigma_j - \lambda)^2} \quad (6.7)$$

nach einer elementaren Rechnung auf

$$R(\lambda) = \frac{S_\sigma(\lambda) - S_1^2(\lambda)}{S_2(\lambda)}, \quad (6.8)$$

und damit ist definiert der Algorithmus

$$\Lambda_{j+1} = \Lambda_j + R(\Lambda_j); \quad j = 1, 2, \dots \quad (6.9)$$

Der Start erfolgt mit einem Hauptwert H_k .

7. Die verkürzte Eigenwertgleichung

Neben der Eigenwertgleichung (6.1) existiert nach [3, Seite 46] die verkürzte Eigenwertgleichung

$$\tilde{f}(\lambda) = S_1(\lambda) - 1 = 0. \quad (7.1)$$

Mit der Ableitung

$$\tilde{f}'(\lambda) = S_1'(\lambda) - 0 = S_2(\lambda) \quad (7.2)$$

lautet somit die Padé-Funktion

$$p_E(\lambda) = \frac{\tilde{f}(\lambda)}{-\tilde{f}'(\lambda)} = \frac{S_1(\lambda) - 1}{-S_2(\lambda)} \quad (7.3)$$

und damit der nach dem Muster von (2.1) abgeleitete Algorithmus

$$\Lambda_{j+1} = \Lambda_j + p_E(\Lambda_j); \quad j = 1, 2, \dots \quad (7.4)$$

Der Start erfolgt mit einem Hauptwert H_k .

8. Die Evolution

8 a) Dieser in [3, Seite 44] eingeführte Algorithmus ist in wenigen Worten beschrieben. Ersetze in der Liste (4.8) die Stützwerte durch die Hauptwerte und fertige mit diesen eine neue Liste an. Dieses Vorgehen wird so oft wiederholt, bis einige oder alle Defekte einen vorgegebenen Schwellwert unterschritten haben. Die Hauptwerte der letzten Liste dienen dann als Startwert für den Algorithmus im Abschnitt 2.

9. Numerische Durchführbarkeit. Ergänzungen

9 a) Ermittlung der Vielfachheit für den Algorithmus (2.9).

Man lässt den Algorithmus für $v = 1, v = 2$ usw. parallel laufen. Konvergenz erfolgt genau einmal. Damit sind Nullstelle und Vielfachheit ermittelt.

Der Taylor-Test mit dem charakteristische Polynom (1.4)

$$\left. \begin{array}{l} v = 1 : \quad \left. \begin{array}{l} f(a) = 0 \\ f'(a) \neq 0 \end{array} \right\} \\ v = 2 : \quad \left. \begin{array}{l} f(a) = 0 \\ f'(a) = 0 \\ f''(a) \neq 0 \end{array} \right\} \end{array} \right\} \quad (9.1)$$

und analog für $v > 2$ kann als Kontrolle dienen.

9 b) Die Matrix (6.2) lautet ausmultipliziert

$$\mathbf{E} = \begin{bmatrix} H_1 & -d_2 & -d_3 & \dots & -d_{m-1} & -d_m \\ -d_1 & H_2 & -d_3 & \dots & -d_{m-1} & -d_m \\ -d_1 & -d_2 & H_3 & \dots & -d_{m-1} & -d_m \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -d_1 & -d_2 & -d_3 & \dots & H_{m-1} & -d_m \\ -d_1 & -d_2 & -d_3 & \dots & -d_{m-1} & H_m \end{bmatrix}, \quad (9.2)$$

und damit wird

$$\text{Spur } \mathbf{E} = \sum_{j=1}^m H_j = \sum_{j=1}^m \lambda_j = -\frac{a_{m-1}}{a_m}, \quad (9.3)$$

womit die Gleichung (5.7) bewiesen ist.

9 c) Einschließungssätze nach Gerschgorin mittels der Matrix

$$\mathbf{F}(\lambda) = \mathbf{E} - \lambda \mathbf{I}_n. \quad (9.4)$$

Es liegt vor ein Kreis mit dem Mittelpunkt

$$H_k = U_k + V_k \cdot i \quad (9.5)$$

und dem Radius

$$r_k = (n-1) \cdot |d_k|. \quad (9.6)$$

Wenn dieser Kreis von den übrigen $n-1$ Kreisen getrennt liegt, so sind zwei Fälle zu unterscheiden.

9 c1) Der Hauptwert H_k ist reell. Dann ist auch der eingeschlossene Eigenwert λ_k reell, und es gilt

$$-r_k + H_k < \lambda_k < H_k + r_k \quad (9.7)$$

9 c2) Für den komplexen Eigenwert

$$\lambda_k = u_k + v_k \cdot i \quad (9.8)$$

dagegen gelten die Einschließungen

$$-r_k + U_k < u_k < U_k + r_k, \quad (9.9)$$

und

$$-r_k + V_k < v_k < V_k + r_k. \quad (9.10)$$

Bei mehrfachen Eigenwerten oder Eigenwerthaufen existieren simultane Einschließungen, siehe dazu [3. Seite 52].

9 d) Ordnungserniedrigung

9 d1) Skalares Polynom. Abspalten einer Nullstelle mittels HORNER-Schema.

9 d2) Matrizenpolynom (1.2). Abspalten von jeweils n Eigenwerten en bloc nach [5].

10. Numerische Beispiele

Erstes Beispiel

Sondierung nach Abschnitt 3 mittels Padé-Funktion $p(\lambda)$ (1.6).

Vorgegeben ist das Polynom

$$f(\lambda) = 4 + 12\lambda + 9\lambda^2 - 4\lambda^3 - 6\lambda^4 + 0 \cdot \lambda^5 + \lambda^6 \quad (10.1)$$

mit den Nullstellen

$$\lambda_1 = \lambda_2 = 2; \quad \lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = -1. \quad (10.2)$$

Sondierung mit $\delta = 0,3$ führt auf die Wertepaare

Λ	$p(\Lambda)$	
3.0000000000000000e-01	-5.261904761904761e-01	}
6.0000000000000000e-01	-9.333333333333332e-01	
9.0000000000000000e-01	-3.483333333333336e+00	
1.2000000000000000e+00	1.466666666666667e+00	
1.5000000000000000e+00	4.166666666666667e-01	
$\lambda_1 = 1.8000000000000000e+00$	1.166666666666663e-01	}
$\lambda_2 = 2.1000000000000000e+00$	-4.696969696969665e-02	

(10.3)

Vorzeichenwechsel

Es folgt die Regula falsi (3.5) gerundet auf

$$\lambda_3 = 2.01389 \quad (10.4)$$

und damit nach (2.9)

$$\left. \begin{array}{c|cc} j & \Lambda_j & p_1(\Lambda_j) \\ \hline 1 & 2.013890000000000e+00 & -3.428770022238466e-03 \\ 2 & 2.006984834339914e+00 & -1.735089829011837e-03 \\ 3 & 2.003502535366870e+00 & -8.728302942998172e-04 \\ 4 & 2.001753817659295e+00 & -4.377505086146021e-04 \\ 5 & 2.000877548907494e+00 & -2.192108705613412e-04 \end{array} \right\} \quad (10.5)$$

Keine Konvergenz für $v = 1$.

$$\left. \begin{array}{c|cc} j & \Lambda_j & p_2(\Lambda_j) \\ \hline 1 & 2.013890000000000e+00 & -6.734450893113345e-03 \\ 2 & 2.000327556690868e+00 & -1.636577281729843e-04 \\ 3 & 2.000000187627338e+00 & -9.381362945114195e-08 \\ 4 & 2.000000000000006e+00 & -3.090806074727201e-14 \\ 5 & 2.000000000000000e+00 & 0 \end{array} \right\} \quad (10.6)$$

Konvergenz für $v = 2$, somit ist

$$\lambda_1 = \lambda_2 = 2. \quad (10.7)$$

Der Ersatz von λ durch $-\lambda$ führt auf das Kopolynom

$$f(-\lambda) = 4 - 12\lambda + 9\lambda^2 + 4\lambda^3 - 6\lambda^4 - 0 \cdot \lambda^5 + \lambda^6. \quad (10.8)$$

Sondierung

$$\left. \begin{array}{cc} \Lambda & p(\Lambda) \\ \hline 3.000000000000000e-01 & -2.064102564102564e-01 \\ 6.000000000000000e-01 & -1.083333333333336e-01 \\ 9.000000000000000e-01 & -2.543859649125041e-02 \\ 1.200000000000000e+00 & 4.848484848484354e-02 \\ 1.500000000000000e+00 & 1.166666666666667e-01 \end{array} \right\} \quad (10.9)$$

Es folgt die Regula falsi (3.5) gerundet auf

$$\lambda_3 = 1.00324 \quad (10.10)$$

und damit der Algorithmus (2.9).

Für $v = 1$, $v = 2$, $v = 3$ tritt keine Konvergenz ein.

Dagegen wird für $v = 4$

$$\left. \begin{array}{c|cc} j & \Lambda_j & p_4(\Lambda_j) \\ \hline 1 & 1.0032400000000000e+00 & -3.191729984318953e-03 \\ 2 & 1.000037928810532e+00 & -3.792209825976858e-05 \\ 3 & 1.000000005273932e+00 & -5.273931558410046e-09 \\ 4 & 1.0000000000000000e+00 & -2.343804163097548e-16 \end{array} \right\} \quad (10.11)$$

Somit vier Mal + 1 im Kopolynom. Folglich

$$\lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = -1 \quad (10.12)$$

MATLAB liefert die Nullstellen

$$\left. \begin{array}{lcl} \tilde{\lambda}_1 & = & 2.0000000000000001e+00 + 7.152216756864169e-09i \\ \tilde{\lambda}_2 & = & 2.0000000000000001e+00 - 7.152216756864169e-09i \\ \tilde{\lambda}_3 & = & -1.000143391292847e+00 \\ \tilde{\lambda}_4 & = & -9.999999991419022e-01 + 1.433904397109860e-04i \\ \tilde{\lambda}_5 & = & -9.999999991419022e-01 - 1.433904397109860e-04i \\ \tilde{\lambda}_6 & = & -9.998566104233441e-01 \end{array} \right\} \quad (10.13)$$

Zweites Beispiel

Diagonaldominante Matrix nach Abschnitt (3.7).

$$\mathbf{F}(\lambda) = \begin{bmatrix} 5+2\lambda+3\lambda^2 & -1 & 0 & 0 & 0 \\ -1 & 9+3\lambda+\lambda^2 & -3 & -2 & 0 \\ 0 & -3 & 6+\lambda^2 & -2 & 0 \\ 0 & -2 & 2 & 12+\lambda+\lambda^2 & -5-\lambda \\ 0 & 0 & 0 & -5-\lambda & 8+4\lambda+4\lambda^2 \end{bmatrix}. \quad (10.14)$$

mit

$$\det \mathbf{F}(\lambda) = f(\lambda) = 12221 + 19366\lambda + 33492\lambda^2 + 28079\lambda^3 + 23637\lambda^4 + 11574\lambda^5 + 5699\lambda^6 + 1631\lambda^7 + 489\lambda^8 + 68\lambda^9 + 12\lambda^{10}. \quad (10.15)$$

Die quadratischen Gleichungen (6.4)

$$\left. \begin{array}{lcl} 5+2\lambda+3\lambda^2 & = & 0 \\ 9+3\lambda+\lambda^2 & = & 0 \\ 6+\lambda^2 & = & 0 \\ 12+\lambda+\lambda^2 & = & 0 \\ 8+4\lambda+4\lambda^2 & = & 0 \end{array} \right\} \quad (10.16)$$

besitzen die Nullstellen

$$\left. \begin{aligned} &-3.33333333333334e-01 + 1.247219128924647e+00i \\ &-3.33333333333334e-01 - 1.247219128924647e+00i \\ &-1.50000000000000e+00 + 2.598076211353316e+00i \\ &-1.50000000000000e+00 - 2.598076211353316e+00i \\ &0 + 2.449489742783178e+00i \\ &0 - 2.449489742783178e+00i \\ &-4.99999999999998e-01 + 3.427827300200522e+00i \\ &-4.99999999999998e-01 - 3.427827300200522e+00i \\ &-5.00000000000000e-01 + 1.32287565532295e+00i \\ &-5.00000000000000e-01 - 1.32287565532295e+00i \end{aligned} \right\} \quad (10.17)$$

Wir wählen die dritte Zeile aus (10.17) als Startwert und erhalten die folgenden Ergebnisse.

a) Algorithmus (2.1).

$$\begin{array}{c|cc} j & \Lambda_j & p(\Lambda_j) \\ \hline 1 & -1.50000000000000e+00+2.598076211353316e+00i & 2.032105570683291e-01-7.395719545396148e-02i \\ 2 & -1.296789442931671e+00+2.524119015899355e+00i & 1.773110756693907e-01-2.275434281511629e-02i \\ 3 & -1.119478367262280e+00+2.501364673084238e+00i & 1.320427191450034e-01+5.627450679466858e-02i \\ 4 & -9.874356481172767e-01+2.557639179878907e+00i & -3.007105982831948e-02+8.920521695652191e-02i \\ 5 & -1.017506707945596e+00+2.646844396835429e+00i & 7.359853863202630e-04-2.137437648438716e-02i \\ 6 & -1.016770722559276e+00+2.625470020351042e+00i & -9.748769601593532e-04-1.081681076639398e-03i \\ 7 & -1.017745599519435e+00+2.624388339274403e+00i & -5.137187053597404e-06+4.029601688399558e-06i \\ 8 & -1.017750736706489e+00+2.624392368876091e+00i & 1.136121179599617e-10-6.578228494562704e-11i \\ 9 & -1.017750736592877e+00+2.624392368810308e+00i & 2.009798662754188e-15-1.502768992132371e-15i \end{array} \quad (10.18)$$

b) Algorithmus (2.9).

$$\begin{array}{c|cc} j & \Lambda_j & p_1(\Lambda_j) \\ \hline 1 & -1.50000000000000e+00+2.598076211353316e+00i & -5.590550503850626e-02-5.178554874365029e-02i \\ 2 & -1.281598940159485e+00+2.530507771744555e+00i & -3.206896765146202e-02-5.435962387848091e-02i \\ 3 & -1.102941934511756e+00+2.519024236220835e+00i & 4.957773050493467e-03-4.558545263614854e-02i \\ 4 & -9.935792103013821e-01+2.581791094008818e+00i & 2.032174795110959e-02+2.087166993263317e-03i \\ 5 & -1.019159105737506e+00+2.632183836150752e+00i & -2.713265716958361e-03+4.879297147422435e-04i \\ 6 & -1.017678176984104e+00+2.624544743975652e+00i & -4.116398540465768e-05+4.357842452009323e-05i \\ 7 & -1.017750658819505e+00+2.624392358442496e+00i & 1.342417115296137e-08+2.442886475494064e-08i \\ 8 & -1.017750736592890e+00+2.624392368810295e+00i & 2.991635783682046e-15-6.560209464970359e-15i \end{array} \quad (10.19)$$

Beide Algorithmen konvergieren quadratisch und liefern nahezu identische Werte. Auf die gleiche Weise werden parallel und unabhängig voneinander die übrigen neun Nullstellen berechnet.

Drittes Beispiel

Evolution nach Abschnitt 8.

Das nach Wilkinson benannte Polynom

$$f(\lambda) = (1 - \lambda)(2 - \lambda) \dots (9 - \lambda)(10 - \lambda) \quad (10.20)$$

oder ausmultipliziert

$$f(\lambda) = 3628800 - 10628640\lambda + 12753576\lambda^2 - 8409500\lambda^3 + 3416930\lambda^4 - 902055\lambda^5 + 157773\lambda^6 - 18150\lambda^7 + 1320\lambda^8 - 55\lambda^9 + \lambda^{10} \quad (10.21)$$

besitzt die Nullstellen 1, 2, . . . , 10.

MATLAB liefert die Näherungen.

$$\left. \begin{aligned} \tilde{\lambda}_1 &= 1.000000000032865e+01 \\ \tilde{\lambda}_2 &= 8.999999998364443e+00 \\ \tilde{\lambda}_3 &= 8.000000003420013e+00 \\ \tilde{\lambda}_4 &= 6.999999996085851e+00 \\ \tilde{\lambda}_5 &= 6.000000002669752e+00 \\ \tilde{\lambda}_6 &= 4.999999998898655e+00 \\ \tilde{\lambda}_7 &= 4.000000000263102e+00 \\ \tilde{\lambda}_8 &= 2.99999999968169e+00 \\ \tilde{\lambda}_9 &= 2.000000000001345e+00 \\ \tilde{\lambda}_{10} &= 1.000000000000000e+00 \end{aligned} \right\} \quad (10.22)$$

Diese benutzen wir als Stützwerte (5.1) und bekommen damit die Liste

$$L_{10} = \left(\begin{array}{ccc} \text{Stützwerte} & \text{Defekte} & \text{Hauptwerte} \\ 1.000000000032865e+01 & 3.727125322099494e-10 & 9.99999999955941e+00 \\ 8.999999998364443e+00 & -1.720094133611112e-09 & 9.000000000084537e+00 \\ 8.000000003420013e+00 & 3.167697847832428e-09 & 8.000000000252316e+00 \\ 6.999999996085851e+00 & -4.044785689387324e-09 & 7.000000000130637e+00 \\ 6.000000002669752e+00 & 2.348194056725277e-09 & 6.000000000321559e+00 \\ 4.999999998898655e+00 & -1.197945997413012e-09 & 5.000000000096601e+00 \\ 4.000000000263102e+00 & 2.777798930869391e-10 & 3.99999999985322e+00 \\ 2.99999999968169e+00 & -3.340011018696152e-11 & 3.000000000001569e+00 \\ 2.000000000001345e+00 & 1.21265960237197e-12 & 2.000000000000132e+00 \\ 1.000000000000000e+00 & 0.000000000000000e+00 & 1.000000000000000e+00 \end{array} \right) \quad (10.23)$$

Es folgen die beiden Evolutionen

$$L_{10} = \left(\begin{array}{ccc} \text{Stützwerte} & \text{Defekte} & \text{Hauptwerte} \\ 9.99999999955941e+00 & -5.567454977060759e-11 & 1.000000000001162e+01 \\ 9.000000000084537e+00 & -3.564064318273009e-11 & 9.000000000120178e+00 \\ 8.000000000252316e+00 & -8.075158037691484e-11 & 8.000000000333067e+00 \\ 7.000000000130637e+00 & 2.082540757136542e-10 & 6.99999999922383e+00 \\ 6.000000000321559e+00 & 2.186021042651543e-10 & 6.000000000102957e+00 \\ 5.000000000096601e+00 & 6.758556180612028e-11 & 5.00000000029015e+00 \\ 3.99999999985322e+00 & -1.347399557668777e-11 & 3.9999999998797e+00 \\ 3.000000000001569e+00 & 4.296279733101633e-12 & 2.99999999997273e+00 \\ 2.000000000000132e+00 & 6.929483440822838e-14 & 2.000000000000063e+00 \\ 1.000000000000000e+00 & 0.000000000000000e+00 & 1.000000000000000e+00 \end{array} \right) \quad (10.24)$$

$$L_{10} = \left(\begin{array}{ccc} \text{Stützwerte} & \text{Defekte} & \text{Hauptwerte} \\ 1.000000000001162e+01 & 1.774461056701600e-11 & 9.99999999993872e+00 \\ 9.000000000120178e+00 & -7.047284661419908e-11 & 9.000000000190651e+00 \\ 8.000000000333067e+00 & -7.797978696197630e-11 & 8.000000000411047e+00 \\ 6.99999999922383e+00 & -3.380356010780448e-10 & 7.000000000260418e+00 \\ 6.000000000102957e+00 & 1.409918897505596e-10 & 5.99999999961965e+00 \\ 5.00000000029015e+00 & 7.793359042933322e-11 & 4.99999999951082e+00 \\ 3.99999999998797e+00 & -5.928558055009481e-12 & 4.00000000004725e+00 \\ 2.99999999997273e+00 & -2.910383045325759e-12 & 3.000000000000184e+00 \\ 2.000000000000063e+00 & 4.619655627634508e-14 & 2.000000000000016e+00 \\ 1.000000000000000e+00 & 0.000000000000000e+00 & 1.000000000000000e+00 \end{array} \right) \quad (10.25)$$

Summenkontrolle (3.7)

$$\text{Soll : } \sum_{j=1}^{10} H_j = -\frac{-55}{a} = 55. \quad \text{Ist: } 49,466 \quad (10.26)$$

Viertes Beispiel

Korrektur von mehrfachen Eigenwerten nach (2.9).

Vorgegeben ist die Matrix

$$\mathbf{F}(\lambda) = \mathbf{A} - \lambda \mathbf{B} \quad (10.27)$$

mit

$$\mathbf{A} = \left[\begin{array}{cccc|c} -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & -1 \end{array} \right]; \quad \mathbf{B} = \mathbf{I}_5. \quad (10.28)$$

Ihre Eigenwerte sind

$$\lambda_1 = -2; \quad \lambda_2 = \lambda_3 = -1; \quad \lambda_4 = \lambda_5 = 1. \quad (10.29)$$

MATLAB liefert die Näherungen

$$\begin{aligned} \tilde{\lambda}_1 &= -1.999999999999996e+00 \\ \tilde{\lambda}_2 &= 1.000000000000000e+00 + 7.768125062636118e-09i \\ \tilde{\lambda}_3 &= 1.000000000000000e+00 - 7.768125062636118e-09i \\ \tilde{\lambda}_4 &= -1.000000009896685e+00 \\ \tilde{\lambda}_5 &= -9.999999901033162e-01 \end{aligned} \quad (10.30)$$

Es folgen die Korrekturen nach (2.9).

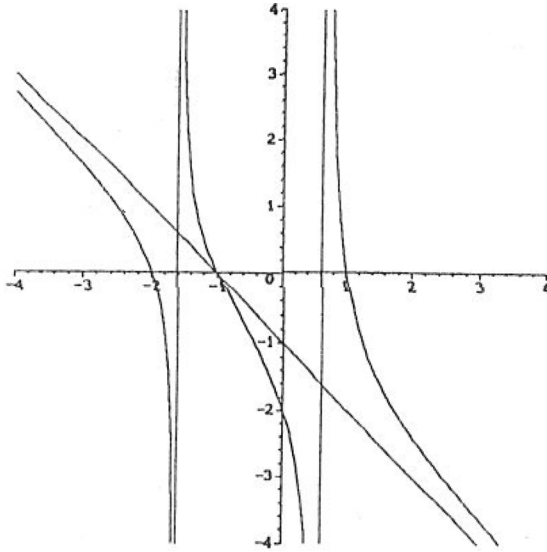


Bild 2: Die Padé-Funktion zur Matrix (10.28)

Start mit $\tilde{\lambda}_5$.

j	Λ_j	$p_1(\Lambda)$	
1	-9.999999901033162e-01	-1.121813169708485e-08	(10.31)
2	-1.000000001321448e+00	0	

Keine Konvergenz für $v = 1$.

j	Λ_j	$p_2(\Lambda)$	
1	-9.999999901033162e-01	-9.896683722532278e-09	(10.32)
2	-9.999999999999999e-01	-1.665334536937735e-16	

Konvergenz für $v = 2$.

Somit $\lambda_4 = \lambda_5 = 1$.

Start mit $\tilde{\lambda}_1$.

j	Λ_j	$p_1(\Lambda)$	
1	-1.999999999999999e+00	-3.552713678800562e-15	(10.33)
2	-2.000000000000000e+00	0	

Somit $\lambda_1 = -2$.

Start mit $\tilde{\lambda}_2$.

$$\begin{array}{c|cc} j & \Lambda_j & p_1(\Lambda) \\ \hline 1 & 1.00000000000000e+00+7.768125062636118e-09i & -1.491714852512857e-16-4.764011116681661e-09i \\ 2 & 1.00000000000000e+00+3.004113945954457e-09i & -2.065119601239563e-16-1.334040524532894e-23i \end{array} \quad (10.34)$$

Keine Konvergenz für $v = 1$.

$$\begin{array}{c|cc} j & \Lambda_j & p_2(\Lambda) \\ \hline 1 & 1.00000000000000e+00+7.768125062636118e-09i & -5.744419753425677e-16-7.768125062636107e-09i \\ 2 & 9.99999999999999e-01+1.158052857574239e-23i & 1.295260195396017e-16-1.158052857574239e-23i \end{array} \quad (10.35)$$

Konvergenz für $v = 2$.

Somit $\lambda_2 = \lambda_3 = -1$.

Fünftes Beispiel

Mehrfache komplexe Nullstellen.

Das Polynom

$$f(\lambda) = (1 + \lambda + \lambda^2)^3 \cdot (1 + \lambda^2)^2 \cdot 6 \quad (10.36)$$

ergibt ausmultipliziert

$$f(\lambda) = 6 + 18\lambda + 48\lambda^2 + 78\lambda^3 + 114\lambda^4 + 120\lambda^5 + 114\lambda^6 + 78\lambda^7 + 48\lambda^8 + 18\lambda^9 + 6\lambda^{10}. \quad (10.37)$$

Die Nullstellen sind

$$\left. \begin{array}{ll} \lambda_1 & = 0 + i \\ \lambda_2 & = 0 - i \\ \lambda_3 & = 0 + i \\ \lambda_4 & = 0 - i \\ \lambda_5 & = -0.5 + \sqrt{0.75}i \\ \lambda_6 & = -0.5 - \sqrt{0.75}i \\ \lambda_7 & = -0.5 + \sqrt{0.75}i \\ \lambda_8 & = -0.5 - \sqrt{0.75}i \\ \lambda_9 & = -0.5 + \sqrt{0.75}i \\ \lambda_{10} & = -0.5 - \sqrt{0.75}i \end{array} \right\} \quad (10.38)$$

mit

$$\sqrt{0.75} = 8.660254037844386e - 01. \quad (10.39)$$

MATLAB liefert die Näherungen

$$\left. \begin{aligned} \tilde{\lambda}_1 &= 2.103940549558203e-08 + 1.0000000028920264e+00i \\ \tilde{\lambda}_2 &= 2.103940549558203e-08 - 1.0000000028920264e+00i \\ \tilde{\lambda}_3 &= -2.103939766850971e-08 + 9.999999710797240e-01i \\ \tilde{\lambda}_4 &= -2.103939766850971e-08 - 9.999999710797240e-01i \\ \tilde{\lambda}_5 &= -5.000094136551562e-01 + 8.660276783463672e-01i \\ \tilde{\lambda}_6 &= -5.000094136551562e-01 - 8.660276783463672e-01i \\ \tilde{\lambda}_7 &= -4.999933232335635e-01 + 8.660324192348879e-01i \\ \tilde{\lambda}_8 &= -4.999933232335635e-01 - 8.660324192348879e-01i \\ \tilde{\lambda}_9 &= -4.999972631112927e-01 + 8.660161137720683e-01i \\ \tilde{\lambda}_{10} &= -4.999972631112927e-01 - 8.660161137720683e-01i \end{aligned} \right\} \quad (10.40)$$

Korrektur von $\tilde{\lambda}_5$ nach (2.9).

$$\begin{array}{c|cc} j & \Lambda_j & p_1(\Lambda_j) \\ \hline 1 & -5.000094136551562e-01+8.660276783463672e-01i & 2.590606103860521e-06+4.667337884496285e-07i \\ 2 & -5.000068230490523e-01+8.660272116125788e-01i & 5.061285460948595e-06-9.728311147477021e-07i \\ 3 & -5.000017617635913e-01+8.660262387814640e-01i & -3.494617894702583e-05+1.077378144656732e-05i \\ 4 & -5.000367079425384e-01+8.660370125629105e-01i & 1.215888288037037e-05-3.935532380990325e-06i \\ 5 & -5.000245490596580e-01+8.660330770305296e-01i & 8.159422114632196e-06-2.552780638469371e-06i \end{array} \quad (10.41)$$

Keine Konvergenz für $v = 1$.

$$\begin{array}{c|cc} j & \Lambda_j & p_2(\Lambda_j) \\ \hline 1 & -5.000094136551562e-01+8.660276783463672e-01i & 4.706686932467359e-06-1.137278373809355e-06i \\ 2 & -5.000047069682237e-01+8.660265410679934e-01i & 2.353445844274929e-06-5.686590462229913e-07i \\ 3 & -5.000023535223794e-01+8.660259724089472e-01i & 1.176760802740233e-06-2.842232846214180e-07i \\ 4 & -5.000011767615767e-01+8.660256881856626e-01i & 5.883348476664755e-07-1.420315348699738e-07i \\ 5 & -5.000005884267291e-01+8.660255461541277e-01i & 2.940921445040524e-07-7.102102632529488e-08i \end{array} \quad (10.42)$$

Keine Konvergenz für $v = 2$.

$$\begin{array}{c|cc} j & \Lambda_j & p_3(\Lambda_j) \\ \hline 2 & -5.000000008552082e-01+8.660254038125426e-01i & 8.552084998311037e-10-2.810326544530710e-11i \\ 3 & -4.99999999999997e-01+8.660254037844393e-01i & -1.553244317250170e-15-2.433903049193858e-15i \end{array} \quad (10.43)$$

Konvergenz für $v = 3$, somit dreifache Nullstelle λ_5 aus (10.38) Das Polynom (10.37) ist (unbeabsichtigt) hermitesch, aber von gerader Ordnung $m = 10$, somit ist -1 keine Nullstelle. Man vergleiche dazu das zehnte Beispiel.

Sechstes Beispiel

Verkürzte Eigenwertgleichung nach (7.3) mit

$$p_E(\lambda) = \frac{S_1(\lambda) - 1}{-S_2(\lambda)} \quad (10.44)$$

Wilkinson-Polynom (10.20)

$$\left(\begin{array}{ccc} \text{Stützwerke} & \text{Defekte} & \text{Hauptwerte} \\ 1.000100000000000e+00 & 9.988178397837826e-05 & 1.000000118216022e+00 \\ 2.000200000000000e+00 & 1.997514392797159e-04 & 2.000000248560720e+00 \\ 3.000300000000000e+00 & 2.996318462367703e-04 & 3.000000368153763e+00 \\ 4.000400000000000e+00 & 3.995415371807089e-04 & 4.000000458462819e+00 \\ 5.000500000000000e+00 & 4.995001173401799e-04 & 5.000000499882660e+00 \\ 6.000599999999999e+00 & 5.995318981743213e-04 & 6.000000468101825e+00 \\ 7.000699999999999e+00 & 6.996716090711247e-04 & 7.000000328390928e+00 \\ 8.000800000000000e+00 & 7.999787558876377e-04 & 8.000000021244112e+00 \\ 9.000900000000000e+00 & 9.005807841077982e-04 & 8.999999419215891e+00 \\ 1.000100000000000e+01 & 1.001930111657591e-03 & 9.999998069888342e+00 \end{array} \right) \quad (10.45)$$

Start mit dem Hauptwert H3

$$\begin{array}{c|cc} j & \Lambda_j & p_E(\Lambda_j) \\ \hline 1 & 3.000000368155010e+00 & -3.677030854107595e-07 \\ 2 & 3.000000000451924e+00 & -4.517926727218366e-10 \\ 3 & 3.000000000000131e+00 & -5.277591961897212e-16 \end{array} \quad (10.46)$$

Zum Vergleich der Algorithmus (2.1)

$$\begin{array}{c|cc} j & \Lambda_j & p(\Lambda_j) \\ \hline 1 & 3.000000368155010e+00 & -3.681554823412160e-07 \\ 2 & 2.999999999999527e+00 & 2.448417482860382e-12 \\ 3 & 3.000000000001975e+00 & -8.777345693336323e-13 \end{array} \quad (10.47)$$

Siebttes Beispiel

Singularäre Leitmatrix nach (4.10).

Vorgegeben ist die Polynommatrix

$$\mathbf{F}(\lambda) = \mathbf{A}_0 + \mathbf{A}_1\lambda + \mathbf{A}_2\lambda^2 + \mathbf{A}_3\lambda^3 + \mathbf{A}_3\lambda^3 + \mathbf{A}_4\lambda^4 \quad (10.48)$$

mit den Koeffizientenmatrizen

$$\mathbf{A}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (10.49)$$

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\mathbf{A}_2 = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix},$$

$$\mathbf{A}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$\mathbf{A}_4 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},$$

wo die Leitmatrix \mathbf{A}_4 regulär ist, somit existieren weniger als $m = \rho \cdot n$ Eigenwerte. Wir unterscheiden zwei Vorgehensweisen.

a) über die Matrix

$$\mathbf{F}(\lambda) = \begin{bmatrix} 1 + \lambda + 2\lambda^2 & \lambda + \lambda^2 + \lambda^4 \\ \lambda & 1 + \lambda + \lambda^2 \end{bmatrix}. \quad (10.50)$$

b) über das charakteristische Polynom

$$\det \mathbf{F}(\lambda) = f(\lambda) = 1 + 2\lambda + 3\lambda^2 + 2\lambda^3 + 2\lambda^4 - \lambda^5 \quad (10.51)$$

vom Grad 5; somit existieren nur 5 Nullstellen und damit auch nur 5 Eigenwerte.

Wir beginnen mit einer Sondierung mittels der Padé-Funktion (1.6) und wählen $\delta = 0.1$:

Λ_j	$p(\Lambda_j)$	
0.0	$-5.000000000000000e-01$	}
0.1	$-4.619268978444236e-01$	
0.2	$-4.401830663615560e-01$	
0.3	$-4.291374155685971e-01$	
0.4	$-4.263509749303621e-01$	
0.5	$-4.304347826086957e-01$	
0.6	$-4.404343891402715e-01$	
0.7	$-4.556680863013057e-01$	
0.8	$-4.756872635561161e-01$	
0.9	$-5.002673598217602e-01$	
1.0	$-5.294117647058824e-01$	
1.1	$-5.633714657980456e-01$	
1.2	$-6.026897069872278e-01$	
1.3	$-6.482868885971897e-01$	
1.4	$-7.016125923546420e-01$	
1.5	$-7.649164677804295e-01$	
1.6	$-8.417448071216616e-01$	
1.7	$-9.379005667657428e-01$	
1.8	$-1.063447440169645e+00$	
1.9	$-1.237357850755909e+00$	
2.0	$-1.500000000000000e+00$	
2.1	$-1.954889850621248e+00$	
2.2	$-2.970956408588159e+00$	
2.3	$-7.548939562102705e+00$	
2.4	$8.638597785977867e+00$	
2.5	$2.306324110671937e+00$	
2.6	$1.151192462987887e+00$	
2.7	$6.571120443038653e-01$	
2.8	$3.772698566917662e-01$	
2.9	$1.934476166846175e-01$	
$\lambda_1 = 3.0$	$6.086956521739131e-02$	} Vorzeichenwechsel
$\lambda_2 = 3.1$	$-4.116006819473399e-02$	

daraus nach der Regula falsi

$$\lambda_3 = 3.059658712064095e+00, \quad (10.53)$$

und weiter nach (3.7) mit $\sigma = 5$

$$\lambda_4 = 3.056811621817845e+00. \quad (10.54)$$

Es folgt der Padé-Algorithmus (2.1)

$$\left. \begin{array}{c|cc} j & \Lambda_j & p(\Lambda_j) \\ \hline 1 & 3.056811621817845e+00 & -2.231403025508382e-06 \\ 2 & 3.056809390414819e+00 & -5.754993999073329e-12 \\ 3 & 3.056809390409065e+00 & -7.589857143243228e-17 \end{array} \right\} \quad (10.55)$$

a) MATLAB liefert mit der Matrix (10.50) die Eigenwerte

$$\left. \begin{array}{lcl} \tilde{\lambda}_1 & = & 3.056809390409061e+00 \\ \tilde{\lambda}_2 & = & -2.519863356616351e-02 + 8.564441482962060e-01i \\ \tilde{\lambda}_3 & = & -2.519863356616351e-02 - 8.564441482962060e-01i \\ \tilde{\lambda}_4 & = & -5.032060616383680e-01 + 4.386303194022314e-01i \\ \tilde{\lambda}_5 & = & -5.032060616383680e-01 - 4.386303194022314e-01i \end{array} \right\} \quad (10.56)$$

b) MATLAB liefert mit dem Polynom (10.51) die Nullstellen

$$\left. \begin{array}{lcl} \tilde{\lambda}_1 & = & 3.056809390409070e+00 \\ \tilde{\lambda}_2 & = & -2.519863356616392e-02 + 8.564441482962062e-01i \\ \tilde{\lambda}_3 & = & -2.519863356616392e-02 - 8.564441482962062e-01i \\ \tilde{\lambda}_4 & = & -5.032060616383678e-01 + 4.386303194022318e-01i \\ \tilde{\lambda}_5 & = & -5.032060616383678e-01 - 4.386303194022318e-01i \end{array} \right\} \quad (10.57)$$

Beide stimmen mit hinreichender Genauigkeit untereinander wie mit Λ_3 aus (10.55) überein.

Achtes Beispiel

Vorgegeben ist das Polynom

$$f(\lambda) = (\lambda - 1)(\lambda - 2)(\lambda - 3)(\lambda - 4)(\lambda - 5) \cdot 3 \quad (10.58)$$

oder ausmultipliziert

$$f(\lambda) = -360 + 822\lambda - 675\lambda^2 + 255\lambda^3 - 45\lambda^4 + 3\lambda^5. \quad (10.59)$$

Die Sondierung mit $\delta = 0.3$ führt auf die Wertepaare

$$\left. \begin{array}{cc} \Lambda_j & p(\Lambda_j) \\ \hline 0.0 & 4.379562043795621e-01 \\ 0.3 & 3.484061594869381e-01 \\ 0.6 & 2.408279034112688e-01 \\ \lambda_1 = 0.9 & 8.366965417990657e-02 \\ \lambda_2 = 1.2 & -3.884787018255549e-01 \end{array} \right\} \quad (10.60)$$

Vorzeichenwechsel

Daraus die Regula falsi

$$\lambda_3 = 9.531631550437540e - 01. \quad (10.61)$$

Wir testen den Algorithmus von Halley nach (2.4).

j	Λ_j	$h(\Lambda_j)$	(10.62)
1	9.000000000000000e-01	1.180808950230746e-01	
2	1.018080895023075e+00	-1.738293306349123e-02	
3	1.000697961959583e+00	-6.969460936557861e-04	
4	1.000001015865928e+00	-1.015863777488736e-06	
5	1.0000000000002150e+00	-2.150576013567233e-12	
6	9.99999999999994e-01	7.894919286223337e-16	
7	1.000000000000000e+00	0	

In der Theorie heißt es nach [6] dazu: die Konvergenz des Halleyschen Algorithmus ist kubisch für einfache, quadratisch für mehrfache Nullstellen, doch ist (zumindest in diesem Beispiel) eine kubische Konvergenz nicht zu erkennen.

Zum Vergleich: die beschleunigte Regula falsi im Anschluss an (10.61) führt in fünf Schritten auf die exakte Lösung.

$$\left. \begin{array}{ll} \lambda_4 = 9.985736255069474e - 01 & ; \quad p(\lambda_4) = 1.422152519655597e - 03 \\ \lambda_5 = 9.999780098758768e - 01 & ; \quad p(\lambda_5) = 2.198911675590514e - 05 \\ \lambda_6 = 1.000000002011310e + 00 & ; \quad p(\lambda_6) = -2.011309654213950e - 09 \\ \lambda_7 = 1.000000000000000e + 00 & ; \quad p(\lambda_7) = 0 \end{array} \right\} \quad (10.63)$$

Literatur

- [1] CARSTENSEN, C. & E. STEIN (1987): Analysis und Berechnung der Falk'schen ECP-Transformation und verwandte Probleme, International Series of Numerical Mathematics, vol. **83**, pp. 47–61.
- [2] CARSTENSEN, C. & E. STEIN (1989): Über die Falksche ECP-Transformation und Verallgemeinerungen. Z. Angew. Math. Mech. **69**, 11, 375–391.
- [3] FALK, S. (2004): Der Eigenwertalgorithmus ECP für Polynommatrizen. Abhandlungen der Braunschweigischen Wissenschaftlichen Gesellschaft Band 54.
- [4] FALK, S. (2011): Die beschleunigte Ritz-Iteration für Polynome und Polynommatrizen, Abhandlungen der Braunschweigischen wissenschaftlichen Gesellschaft Band 58.
- [5] FALK, S. (1994): Das Reduktionsverfahren für Polynommatrizen, ZAMM **74**/1, pp. 3–15.
- [6] MATHIS, W., T. THIESSEN & S. FALK: Iterationsverfahren für Nullstellenprobleme. Institut für Theoretische Elektrotechnik, Leibniz Universität Hannover.

- [7] SHAW, M. & J.F. TRAUB (1974): On the number of mulitplications for the evaluation of a polynomial and some of its derivatives. J. Assoc. Comp. Mech **21**, pp. 61–166.
- [8] STEWART, G.W. (1968): Some topics in numerical analysis, I Lehmer's method for finding the zeros of a polynomial. Technical report, Oak Ridge.
- [9] ZURMÜHL, R. & S. FALK (1992): Matrizen und ihre Anwendungen 1, Grundlagen. 6. Auflage Springer-Verlag Berlin.
- [10] ZURMÜHL, R. & S. FALK (1986): Matrizen und ihre Anwendungen 2, Numerische Methoden. 5. Auflage Springer-Verlag Berlin.

Mein Dank für die erfolgreiche Zusammenarbeit geht an die Herren M.Sc. Jonas Denißen, Berlin und an Prof. Dr.-Ing. Wolfgang Mathis, Leibniz Universität Hannover.